


Федеральное государственное автономное
образовательное учреждение
высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»
ИНСТИТУТ КОСМИЧЕСКИХ И ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ
Кафедра «Информатика»

УТВЕРЖДАЮ

Заведующий кафедрой

 Рубан А.И.


«21» июня 2016 г.

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

23100.62 «Программная инженерия»

Личный кабинет студента СФУ ИКИТ

Руководитель

 21.06.16
подпись, дата

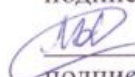
доцент, к.т.н.

должность, ученная степень

Ю. Ю. Якунин

инициалы, фамилия


Выпускник


подпись, дата

М. О. Димухаметов

инициалы, фамилия

Нормконтроль


подпись, дата

доцент, к.т.н.

должность, ученная степень

О. А. Антамошкин

инициалы, фамилия

Красноярск 2016 г.

Федеральное государственное автономное
образовательное учреждение
высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»
ИНСТИТУТ КОСМИЧЕСКИХ И ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ
Кафедра «Информатика»

УТВЕРЖДАЮ

Заведующий кафедрой

_____/Рубан А.И.

« ____ » _____ 2016 г.

**ЗАДАНИЕ
НА ВЫПУСКНУЮ КВАЛИФИКАЦИОННУЮ РАБОТУ
в форме бакалаврской работы**

Студенту Димухаметову Максиму Олеговичу
Группа КИ12-18Б Направление (специальность): 23100.62, Программная инженерия.

Тема выпускной квалификационной работы: “Личный кабинет студента СФУ ИКИТ”

Утверждена приказом по университету № 6145/с от 10.05.2016.

Руководитель ВКР: Ю. Ю. Якунин, доцент кафедры “Информатика”, кандидат технических наук.

Исходные данные для ВКР: создание личного кабинета студента СФУ ИКИТ на основании приказа №1185 от 10.10.2014г. “О развитии электронного института”; создание электронной зачетной книжки; создание модуля опросов студентов по учебно-организационным вопросам.

Перечень разделов ВКР:

- введение;
- анализ предметной области;
- технологическая платформа приложения;
- описание приложения;
- заключение.

Перечень графического материала: презентационные слайды PowerPoint.

Руководитель ВКР

подпись

Ю. Ю. Якунин
инициалы, фамилия

Задание принял к исполнению

подпись

М. О. Димухаметов
инициалы, фамилия

« ____ » _____ 2016 г.

РЕФЕРАТ

Выпускная квалификационная работа на тему “Личный кабинет студента СФУ ИКИТ” содержит 51 страницу текстового документа, 15 рисунков, 2 таблицы и 13 использованных источников.

ЛИЧНЫЙ КАБИНЕТ СТУДЕНТА, СТУДЕНТ, СИСТЕМА.

Цель - создание и внедрение подсистемы Личного кабинета студента в ИКИТ СФУ.

В квалификационной работе была решена задача создания личного кабинета студента в Институте космических и информационных технологий (ИКИТ) Сибирского федерального университета (СФУ). Решением данной системы является подсистема автоматизированной системы управления (АСУ) ИКИТ.

Так как данная система будет внедряться уже в эксплуатирующую ряд систем на базе ИКИТ, то был произведен обзор данных систем, и был разработан информационный интеграционный инструмент.

Содержание

Введение

1. Анализ предметной области

1.1 Обзор существующих решений

1.2 Анализ существующих программных решений

1.3 Функциональные и нефункциональные требования

2. Технологическая платформа приложения

2.1 Язык программирования

2.2 Фреймворки

2.3 СУБД

2.4 Система версионного управления

2.5 Интегрированная среда разработки

2.6 Система сборки проектов

3. Описание приложения

3.1 Архитектура взаимодействия системы

3.2 Архитектура компонентов системы

3.3 Архитектура пакетов системы

3.4 Описание функций

Заключение

Список литературы

Приложения

Введение

Применение информационных технологий обеспечивает автоматизацию многих процессов высшего учебного заведения. Одним из направлений автоматизации в высшем учебном заведении - является создание личного кабинета студента. На сегодняшний день в Институте космических и информационных технологий (ИКИТ) действует автоматизированная система управления (АСУ) ИКИТ, которая предназначена для повышения качества работы учебно-организационного отдела (УОО), а также для избавления от рутинной работы с бумагами.

Анализируя функции АСУ ИКИТ, работу УОО и преподавателей, можно заметить, что на данный момент студент не участвует в работе автоматизации системы. Для решения этой проблемы была поставлена цель.

Целью квалификационной работы является создание и внедрение подсистемы Личного кабинета студента в ИКИТ СФУ.

Для достижения поставленной цели, нужно решить следующие задачи:

1. Обзор имеющихся систем.
2. Разработать архитектуру личного кабинета студента.
3. Разработать Личного кабинета студента и внедрить в эксплуатацию.

1. Анализ предметной области

Если рассматривать университет, со стороны студента, то для него главными документами являются зачетная книжка и журнал старосты. Именно эти документы нужно перевести в электронный вид. Перед тем как начать разрабатывать собственное решение, необходимо исследовать имеющиеся технологии на рынке.

1.1 Обзор существующих решений

Существует множество программного обеспечения, реализующая функциональность личного кабинета студента. Чтобы выбрать наиболее подходящее был произведен анализ существующих решений.

1.1.1 По способам распространения

Платные

Платные продукты делятся на два типа:

- разовая покупка, после оплаты предоставляется доступ к программному обеспечению на постоянной основе;
- покупка на период лицензированного ключа, который нужно обновлять по истечению указанного периода.

Бесплатные

Бесплатные продукты бывают двух типов:

- с открытым исходным кодом, которые можно использовать из открытых источников, например github;

- ограниченная лицензия. Подобные можно использовать только с указанием ссылки на автора.

Бесплатные с ограниченной функциональностью

Это такой продукт, когда доступна бесплатная часть с ограниченной функциональностью, но есть возможность приобрести дополнительный функционал за определенную стоимость, которая также как и платные системы делится на два типа: разовая покупка и покупка на период.

1.1.2 По предоставляемому функционалу

Перечень программ можно разделить по функциям. Перед тем, как начать собирать нужный набор программ, необходимо определить задачи для будущей системы, а именно какие данные нужны и как пользователь будет оперировать ими. В данном случае необходимо собрать информацию по студентам, получить отчет по ним и провести подписание данных отчетов.

1.1.3 Обзор программного обеспечения функционирующего в ИКИТ

На данный момент, в ИКИТ СФУ функционируют следующие системы: Сервер LDAP, информационная система (ИС) “Деканат” города Шахт, АИС “Абитуриент” и АСУ ИКИТ

Сервер LDAP

LDAP (англ. Lightweight Directory Access Protocol — «облегчённый протокол доступа к каталогам») — протокол прикладного уровня для доступа к службе каталогов X.500, разработанный IETF как облегчённый вариант

разработанного ITU-T протокола DAP. LDAP — относительно простой протокол, использующий TCP/IP и позволяющий производить операции аутентификации (bind), поиска (search) и сравнения (compare), а также операции добавления, изменения или удаления записей. Обычно LDAP-сервер принимает входящие соединения на порт 389 по протоколам TCP или UDP. Для LDAP-сеансов, инкапсулированных в SSL, обычно используется порт 636.

Всякая запись в каталоге LDAP состоит из одного или нескольких атрибутов и обладает уникальным именем (DN — англ. Distinguished Name). Уникальное имя может выглядеть, например, следующим образом: «cn=Иван Петров, ou=Сотрудники, dc=example, dc=com». Уникальное имя состоит из одного или нескольких относительных уникальных имен (RDN — англ. Relative Distinguished Name), разделённых запятой. Относительное уникальное имя имеет вид ИмяАтрибута=значение. На одном уровне каталога не может существовать двух записей с одинаковыми относительными уникальными именами. В силу такой структуры уникального имени записи в каталоге LDAP можно легко представить в виде дерева.

Запись может состоять только из тех атрибутов, которые определены в описании класса записи (*object class*), которые, в свою очередь, объединены в схемы (*schema*). В схеме определено, какие атрибуты являются для данного класса обязательными, а какие — необязательными. Также схема определяет тип и правила сравнения атрибутов. Каждый атрибут записи может хранить несколько значений.[1]

Логин и пароль для сервера СФУ LDAP имеется у каждого студента и сотрудника СФУ, поэтому именно через эту систему будет производиться аутентификация студентов.

ИС “Деканат”

Информационная система, разработанная лабораторией математического моделирования и информационных систем (ММИС) г. Шахт.

Информационная система “Деканат” предназначена для ведения личных дел студентов и может работать отдельно или в составе ИС “Электронные ведомости”. Она позволяет автоматизировать:

- управление учебными группами и специальностями, включая создание отдельных списков групп на каждый учебный год;
- создание электронных личных дел студентов;
- перевод студентов в другую группу, зачисление, отчисление и восстановление, перевод в академический отпуск и т.д.;
- импорт списка групп, сформированного из ИС “Планы”;
- зачисление студентов из ИС “Приемная комиссия” и распределение по учебным группам;
- расчет итогового рейтинга студентов за определенный период времени;
- поиск студентов в базе данных;
- получение сводных данных по контингенту студентов и формирование отчетов;
- склонение ФИО студентов по падежам;
- получение статистики для подготовки отчетов 3-НК;
- создание собственных отчетов в Microsoft Office и добавление их в программу.

Программа наглядно отображает структуру образовательного учреждения и обеспечивает удобную навигацию по ее разделам. Часто используемые команды вынесены на панель инструментов.

Для повышения гибкости и учета требований образовательного учреждения программа поддерживает создание пользователями отчетов в

Microsoft Word и Excel. Также в готовых отчетах можно изменять структуру отчета, форматирование текста, добавлять или удалять атрибуты отчета.

Среди часто используемых отчетов, входящих в программу, такие как:

- справка студента;
- справка для военкомата;
- личная карточка студента;
- список группы;
- экзаменационная ведомость;
- зачетная ведомость.

Программа позволяет осуществлять перевод одного или нескольких студентов одновременно из группы в группу, проводить отчисление и восстановление с хранением атрибутов приказов в истории перемещения студента.

Система безопасности построена на базе ролей, которые могут назначаться пользователю для предоставления доступа к заданным факультетам. Пользовательский интерфейс формируется в соответствии с выданными пользователю правами.

ИС “Деканат” интегрируется с системами “Планы”, “Приемная комиссия”, “Электронные ведомости” и “Тестирование”. Для быстрого создания базы студентов поддерживается импорт списков студентов из текстовых или табличных документов.[2]

АИС “Абитуриент”

Система «Абитуриент» - это уникальная автоматизированная электронная база данных по всем абитуриентам СФУ, специально разработанная для проведения приемной кампании Сибирского федерального университета. Став абитуриентом СФУ, есть возможность, отслеживать свое положение в конкурсе

по мере поступления в приемную комиссию сведений о результатах ЕГЭ и прохождения вступительных испытаний.[3]

АИС “Студенческий офис”

Информационная система, разработанная на языке Delphi, функционирующая в ФГАОУ ВПО «Сибирский федеральный университет» на базе Института космических и информационных технологий.

Электронный деканат был разработан в 2007 году, и поддерживался одним человеком, в настоящий момент этот сотрудник уволился, и поддерживать программное обеспечение не кому.

Помимо этого, информационная система Электронный Деканат ИКИТ имеет ряд недостатков:

- отсутствие web интерфейса;
- громоздкая архитектура;
- трудоёмкая поддержка системы;
- отсутствие возможности добавления и использования новых технологий.

На данный момент функционирует, но со всеми отведенными ей задачами не справляется.

АСУ ИКИТ

Программный продукт, разрабатываемый сотрудниками ИКИТ СФУ, является сложной интеграционной системой и состоит из множества разнородных элементов. Данная система является веб-разработкой, функциональность которой почти достигает АИС “Студенческий офис”, а в дальнейшем планирует полностью заменить эту систему.

На данный момент АСУ ИКИТ обладает функциональностью:

- редактор приказов;

- кабинет преподавателя;
- паспорт студента;
- паспорт группы;
- передачи;
- электронно-цифровая подпись (ЭЦП) для ведомостей;
- отчеты по успеваемости;
- ведомости.

1.2 Анализ существующих программных решений

Для выявления необходимой функциональности был произведен сравнительный анализ с аналогами разрабатываемой системы.

Личный кабинет студента АлтГТУ - система, разрабатываемая отделом единой автоматизированной информационной системы Алтайского государственного технического университета (АлтГТУ).

Личный кабинет студента ИрГУПС - система, разрабатываемая в Иркутском государственном университете путей сообщения (ИрГУПС).

Сравнительный анализ систем по функциональности, представлен в таблице 1.

Таблица 1 - сравнительный анализ

Показатель	Личный кабинет студента АлтГТУ	Личный кабинет студента ИрГУПС.
расписание занятий	есть	есть
расписание экзаменов	есть	есть
результаты сессий	есть	есть
системе обмена сообщений с одноклассниками	есть	нет
текущая информация о студенте согласно данным деканата	нет	есть

модуль, опрашивающий студентов по учебным вопросам	нет	нет
оповещения о пересдачах	нет	есть
электронный журнал старосты	нет	нет
модуль посещаемости	нет	нет

Анализ представленных систем позволяет сделать следующий вывод, что разрабатываемая система должна иметь следующие функции:

- расписание занятий;
- электронная зачетная книжка;
- модуль посещаемости;
- журнал старосты;
- оповещения о студентах;
- модуль, опрашивающий студентов.

1.3 Функциональные и функциональные требования

1.3.1 Общее описание

Создание личного кабинета студента предназначено для автоматизации работы некоторых отделов УОО и упрощения работы студентов с деканатом через модули: “Зачетная книжка”, “Журнал старосты”, “Посещаемость”, “Оповещения” и “Опросник”.

Так как сейчас происходит автоматизация учебных процессов, разрабатываемая система личного кабинет студента будет как раз кстати. Выбранная архитектура будет построена по принципу клиент-сервер.

1.3.2 Общие требования

Требование к структуре и функционированию

Личный кабинет студента должна состоять из модулей:

- электронная зачетная книжка;
- журнал старосты;
- оповещения;
- опросник;
- посещаемость.

Система должна функционировать в виде “Клиент-сервер”. Под клиентом будут пониматься студенты, а сервером будет выступать разрабатываемая система. Каждый модуль будет обмениваться данными через система управления базами данных (СУБД) PostgreSQL и СУБД Intersystems Cache.

Требования к эргономике и технической эстетике

Интерфейс` системы рассчитывается создавать понятным пользователю, преследуя стиль известных и современных систем. Стиль должен быть соблюден в корпоративном стиле СФУ.

Требования к защите информации от несанкционированного доступа

В системе не должно присутствовать несанкционированного входа с наличием административных прав. Подключение к базе данных (БД) не должно быть доступно извне. Требуется разработка архитектуры таким способом, чтобы исключить возможность SQL-инъекций.

Дополнительные требования

Дополнительным требованиям служит добавление справки о модулях системы, с указанием подробной инструкции как пользоваться ими.

1.3.3 Требования к функциям, выполняемые системой

Перечень функций:

- отображение зачетной книжки студента;
- показ посещаемости студента;
- занесение информации о посещаемости студентов через модуль “журнал старосты”;
- просмотр уведомлений, отправленных с УОО;
- прохождение опроса по учебно-организационным вопросам.

1.3.4 Требования к информационному обеспечению

1.3.4.1 По составу и структуре данных в системе

- СУБД PostgreSQL хранит информацию по учебно-организационному процессу.
- СУБД Intersystems Cache используется для построения системы уведомлений, опросов и как аналитический инструмент для обработки результатов опросов и построения статистических отчетов.

1.3.4.2 По контролю, хранению, обновлению и восстановлению данных

Резервное копирование данных происходят ежедневно и после каждого изменения структуры базы данных.

1.3.5 Требования к характеристикам взаимосвязей создаваемой системы со смежными системами

Все компоненты должны быть связаны с системой АСУ ИКИТ через СУБД PostgreSQL. В АСУ ИКИТ заполняются данные об оценках, информации о студенте, в свою очередь с АИС “Личный кабинет студента” поступает информацию о посещаемости студентов.

Также должна присутствовать связь через REST сервис, являющийся интерфейсом для базы данных, хранящей данные об уведомлениях и опросах. В качестве REST сервиса будет выступать веб-приложение в Intersystems Ensemble. Информация, заполняемая сотрудниками УОО в системе АСУ ИКИТ, обрабатывается с помощью REST сервиса и предоставляется в АИС “Личный кабинет студента”.

Описанные выше взаимодействия, представлены на рисунке 1.

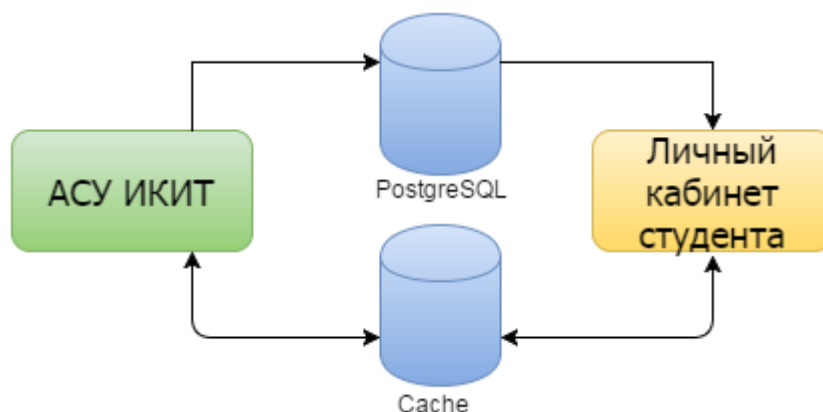


Рисунок 1 - Взаимодействие личный кабинет студента

1.3.6 Требования к архитектуре

Архитектура должна быть спроектирована таким способом, чтобы отвечать следующим требованиям:

- тестируемость (возможность установления факта правильного функционирования);
- диагностируемость (возможность нахождения неисправной части системы);

- безопасность (соответствие требованиям промышленной безопасности и технике безопасности);
- модифицируемость (возможность перенастройки для работы с другими технологическими процессами);
- функциональная расширяемость (возможность ввода в систему дополнительных функциональных возможностей, не предусмотренных в техническом задании);
- масштабируемость (возможность увеличения размера автоматизированной системы при увеличении размера объекта автоматизации);

1.3.7 Требования к интерфейсу

Интерфейс должен быть построен так, чтобы были обеспечены следующие принципы:

- наличие широкого набора иерархических меню, системы подсказок, обучения и т.п.;
- удобный переход между модулями системы;
- возможность сохранения результатов в автоматическом режиме, при ответе на опрос.

2. Технологическая платформа приложения

2.1 Язык программирования

Для создания подсистемы был выбран язык программирования Java, потому что: во-первых, на нем написана АСУ ИКИТ; во-вторых, является самым популярным языком на текущий момент.

Java — объектно-ориентированный язык программирования, разрабатываемый компанией Sun Microsystems с 1991 года и официально выпущенный 23 мая 1995 года. Изначально новый язык программирования назывался Oak (James Gosling) и разрабатывался для бытовой электроники, но впоследствии был переименован в Java и стал использоваться для написания апплетов, приложений и серверного программного обеспечения.

Программы на Java могут быть транслированы в байт-код, выполняемый на виртуальной java-машине (JVM) — программе, обрабатывающей байт-код и передающей инструкции оборудованию, как интерпретатор, но с тем отличием, что байт-код, в отличие от текста, обрабатывается значительно быстрее.[4]

2.2 Фреймворки

ZK — фреймворк для разработки веб-приложений тайваньской компании Potix, реализован полностью на Java. Распространяется под коммерческой лицензией, ограниченная по функциональным возможностям версия доступна свободно под лицензией LGPL. Основные принципы — предоставление возможности разработчику веб-приложений создавать полноценные пользовательские интерфейсы для веб-приложений без применения JavaScript, в событийно-ориентированной парадигме и компонентной модели, что наиболее привычно разработчикам настольных приложений в RAD-средах.

Для описания веб-интерфейса, работающего на стороне клиента, используется специальный, основанный на XML язык разметки ZUML,

максимально близкий к разработанному Mozilla языку XUL. На основании ZUML-описания фреймворк генерирует разметку и JavaScript-код, асинхронно вызывающий серверные методы обработки, создаваемые на Java. Вся обработка событий и контекст пользовательской сессии концентрируется на сервере, в этом смысле ZK относят к «серверо-центричным» Java-фреймворкам, в противовес реализациям, существенно использующих модифицируемую логику на JavaScript на стороне браузера. [5]

Hibernate — библиотека для языка программирования Java, предназначенная для решения задач объектно-реляционного отображения (object-relational mapping — ORM). Она представляет собой свободное программное обеспечение с открытым исходным кодом (open source), распространяемое на условиях GNU Lesser General Public License. Данная библиотека предоставляет легкий в использовании каркас (фреймворк) для отображения объектно-ориентированной модели данных в традиционные реляционные базы данных.

Целью Hibernate является освобождение разработчика от значительного объёма сравнительно низкоуровневого программирования по обеспечению хранения объектов в реляционной базе данных. Разработчик может использовать Hibernate как в процессе проектирования системы классов и таблиц «с нуля», так и для работы с уже существующей базой данных.

Hibernate не только решает задачу связи классов Java с таблицами базы данных (и типов данных Java с типами данных SQL), но и также предоставляет средства для автоматической генерации и обновления набора таблиц, построения запросов и обработки полученных данных и может значительно уменьшить время разработки, которое обычно тратится на ручное написание SQL- и JDBC-кода. Hibernate автоматизирует генерацию SQL-запросов и освобождает разработчика от ручной обработки результирующего набора данных и преобразования объектов, максимально облегчая перенос (портирование) приложения на любые базы данных SQL.

Hibernate обеспечивает прозрачную поддержку сохранности данных (persistence) для «РОЮ» (то есть для стандартных Java-объектов); единственное строгое требование для сохраняемого класса — наличие конструктора по умолчанию (без параметров). Для корректного поведения в некоторых приложениях требуется также уделить внимание методам equals() и hashCode().

Mapping (сопоставление, проецирование) Java-классов с таблицами БД осуществляется с помощью конфигурационных XML-файлов или Java-аннотаций. При использовании файла XML Hibernate может генерировать скелет исходного кода для классов длительного хранения (persistent). В этом нет необходимости, если используется аннотация. Hibernate может использовать файл XML или аннотации для поддержки схемы базы данных.

Обеспечиваются возможности по организации отношения между классами «один-ко-многим» и «многие-ко-многим». В дополнение к управлению связями между объектами Hibernate также может управлять рефлексивными отношениями, где объект имеет связь «один-ко-многим» с другими экземплярами своего собственного типа данных.

Hibernate поддерживает отображение пользовательских типов значений. Это делает возможными такие сценарии:

- Переопределение типа по умолчанию SQL, Hibernate выбирает при отображении столбца свойства.
- Проецирование перечисляемого типа Java на поле БД, будто они являются обычными свойствами.
- Проецирование одного свойства в несколько колонок.[6]

2.3 СУБД

СУБД - совокупность программных и лингвистических средств общего или специального назначения, обеспечивающих управление созданием и использованием баз данных.[7]

PostgreSQL - это объектно-реляционная система управления базами данных, основанная на POSTGRES, версии 4.2, которая была разработана в Научном Компьютерном Департаменте Беркли Калифорнийского Университета. POSTGRES является пионером во многих аспектах, которые стали доступны в некоторых коммерческих СУБД много позже.

PostgreSQL поддерживает большую часть стандарта SQL и предлагает множество современных возможностей:

- комплексные запросы;
- внешние ключи;
- триггеры;
- представления (views);
- транзакционная целостность;
- многоверсионное управление параллельным доступом.

Также, возможности PostgreSQL могут быть расширены пользователем путём добавления новых:

- типов данных;
- функций;
- операторов;
- агрегатных функций;
- индексных методов;
- процедурных языков.

Поскольку СУБД PostgreSQL выпускается под либеральной лицензией, её можно бесплатно использовать, модифицировать и распространять для любых целей, включая личные, коммерческие или академические.[8]

InterSystems Cache – это постреляционная объектная СУБД, предоставляющая практически неограниченные возможности для разработки Web-решений и клиент-серверных приложений. СУБД Cache предназначена для обработки транзакций в системах со сверхбольшими базами данных и

практически неограниченным количеством одновременно работающих пользователей.

Платформа данных InterSystems Caché поддерживает различные типы данных (структурированные и неструктурированные, централизованные и децентрализованные), SQL и NoSQL-способы представления данных, а также выполняет автоматическое управление метаданными. Она обеспечивает внушительное горизонтальное и вертикальное масштабирование. Уникальность продукта состоит в том, что платформа Caché позволяет осуществлять аналитическую обработку первичной информации, гарантируя непрерывность цикла работы с данными: их сбор, распространение, анализ и действия пользователей.[9]

2.4 Система версионного управления

Subversion (SVN) — свободная централизованная система управления версиями, официально выпущенная в 2004 году компанией CollabNet.

Subversion — централизованная система, то есть данные хранятся в едином хранилище. Хранилище может располагаться на локальном диске или на сетевом сервере.

Subversion предлагает два варианта организации репозитория. Репозитории первого типа используют для хранения базы данных на основе Berkeley DB, репозитории второго типа — обычные файлы специального формата. Разработчики Subversion часто называют хранилище «файловой системой», поэтому второй тип получил название FSFS, то есть файловая система (англ. File System) поверх (обычной) файловой системы.[10]

2.5 Интегрированная среда разработки

Интегрированная среда разработки, ИСР/IDE (англ. Integrated development environment) — комплекс программных средств, используемый программистами для разработки программного обеспечения (ПО).

Среда разработки включает в себя:

- текстовый редактор,
- компилятор и/или интерпретатор,
- средства автоматизации сборки,
- отладчик.

Иногда содержит также средства для интеграции с системами управления версиями и разнообразные инструменты для упрощения конструирования графического интерфейса пользователя.[11]

JetBrains IntelliJ IDEA – это ведущая среда быстрой разработки на языке Java. IntelliJ IDEA представляет собой высокотехнологичный комплекс тесно интегрированных инструментов программирования, включающий интеллектуальный редактор исходных текстов с развитыми средствами автоматизации, мощные инструменты рефакторинга кода, встроенную поддержку технологий J2EE, механизмы интеграции со средой тестирования Ant/JUnit и системами управления версиями, уникальный инструмент оптимизации и проверки кода Code Inspection, а также инновационный визуальный конструктор графических интерфейсов. Уникальные возможности JetBrains IntelliJ IDEA избавляют программиста от груза рутинной работы, помогают своевременно устранить ошибки и повысить качество кода, поднимая продуктивность разработчика на новую высоту.

Ключевые возможности:

- Умное автодополнение, инструменты для анализа качества кода, удобная навигация, расширенные рефакторинги и форматирование для Java,

Groovy, Scala, HTML, CSS, javascript, CoffeeScript, ActionScript, LESS, XML и многих других языков.

- Поддержка всех популярных фреймворков и платформ, включая Java EE, Spring Framework, Grails, Play Framework, GWT, Struts, Node.js, AngularJS, Android, Flex, AIR Mobile и многих других.

- Интеграция с серверами приложений, включая Tomcat, TomEE, GlassFish, JBoss, WebLogic, WebSphere, Geronimo, Resin, Jetty и Virgo.

- Инструменты для работы с базами данных и SQL файлами, включая удобный клиент и редактор для схемы базы данных.

- Интеграция с коммерческими системами управления версиями Perforce, Team Foundation Server, ClearCase, Visual SourceSafe.

- Инструменты для запуска тестов и анализа покрытия кода, включая поддержку всех популярных фреймворков для тестирования.[12]

2.6 Система сборки проектов

Apache Maven - это инструмент для сборки Java проекта: компиляции, создания jar, создания дистрибутива программы, генерации документации. Простые проекты можно собрать в командной строке. Если собирать большие проекты с командной строки, то команда для сборки будет очень длинной, поэтому её иногда записывают в bat/sh скрипт. Но такие скрипты зависят от платформы. Для того чтобы избавиться от этой зависимости и упростить написание скрипта используют инструменты для сборки проекта.

Преимущества Maven:

- Независимость от OS. Сборка проекта происходит в любой операционной системе. Файл проекта один и тот же.
- Управление зависимостями. Редко какие проекты пишутся без использования сторонних библиотек(зависимостей). Эти сторонние библиотеки зачастую тоже в свою очередь используют библиотеки

разных версий. Мавен позволяет управлять такими сложными зависимостями. Что позволяет разрешать конфликты версий и в случае необходимости легко переходить на новые версии библиотек.

- Возможна сборка из командной строки. Такое часто необходимо для автоматической сборки проекта на сервере (Continuous Integration).
- Хорошая интеграция со средами разработки. Основные среды разработки на java легко открывают проекты которые собираются с помощью maven. При этом зачастую проект настраивать не нужно - он сразу готов к дальнейшей разработке.
- Как следствие - если с проектом работают в разных средах разработки, то maven удобный способ хранения настроек. Настроечный файл среды разработки и для сборки один и тот же - меньше дублирования данных и соответственно ошибок.
- Декларативное описание проекта.[13]

3. Описание приложения

3.1 Архитектура взаимодействия системы

На общую архитектуру системы АСУ ИКИТ, чьей подсистемой является личный кабинет студента, повлияли эксплуатируемые в СФУ программные средства:

- АИС “Студенческий офис”;
- система единой точки авторизации (LDAP);
- АИС “Абитуриент”;
- система электронного обучения СФУ;
- ИС “Деканат”.

Разрабатываемый система используют всю информацию, собранную в СУБД PostgreSQL, в которой хранится информация об оценках и посещаемости студентов, и СУБД InterSystems Cache, в которой хранится информация по оповещениям и опросам.

Архитектуру взаимодействия представлена на рисунке 2.

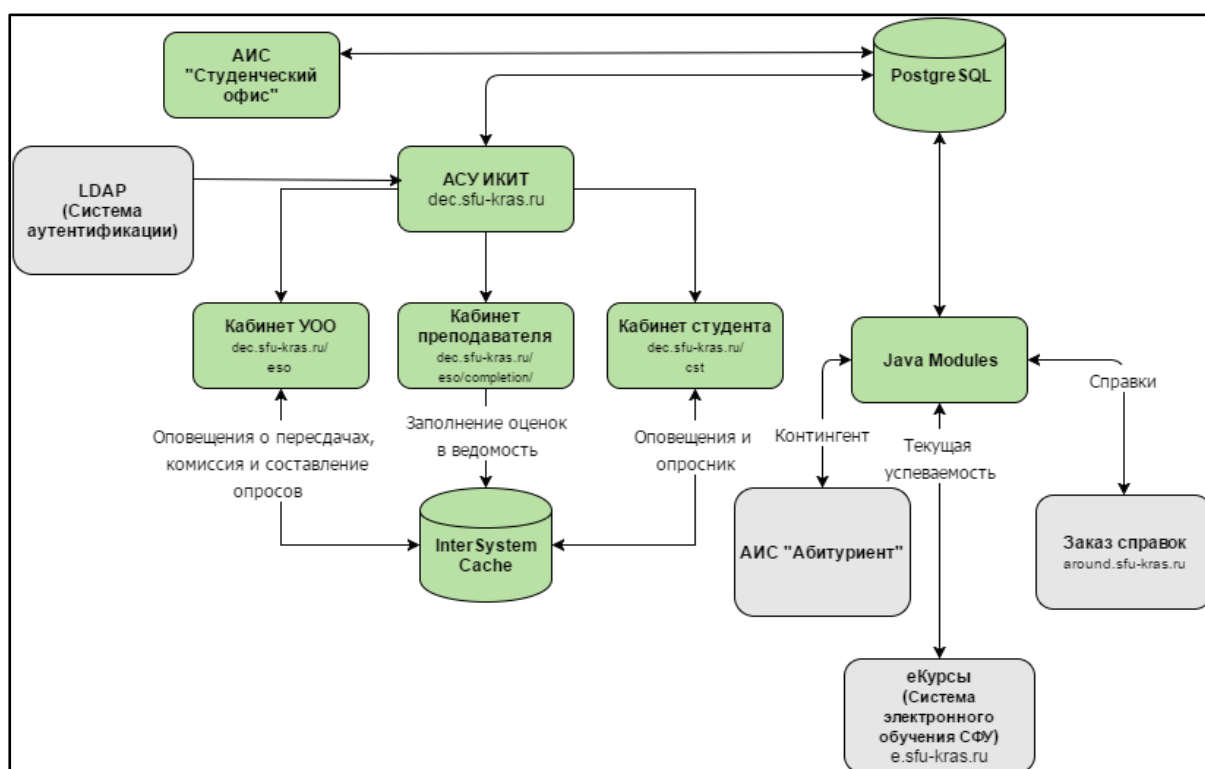


Рисунок 2 - Архитектура взаимодействия системы

3.2 Архитектура пакетов системы

Архитектура пакетов должна быть интуитивно понятна, поэтому изначально главная папка была разделена на 3 раздела:

- resources - папка, где располагаются все конфигурационные файлы, например конфигурационный xml-файла для настройки подключения к БД.
- webapp - в данной папке располагается каталог WEB-INF, внутри которого содержится файл, который является дескриптором развертывания приложения с названием web.xml. Также в папке webapp располагаются zul файлы. Zul - это веб-файл основанный на языке разметки XML.
- java - содержимое данной папки, представлено на рисунке 3.

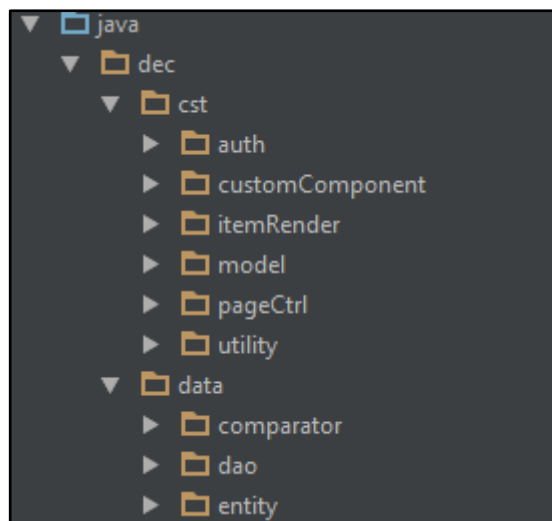


Рисунок 3 - структура папок java

- cst
 - auth - в данной папке располагается настройки авторизации с сервером LDAP;
 - customComponent - компоненты, которые собрали из стандартных, для удобства разработки;

- `itemRender` - в данном каталоге располагается классы, которые отрисовывают такие сложные компоненты, как таблица, выпадающие списки;
- `model` - в данной папке располагаются проху-классы, которые получаются из многотабличных SQL-запросов;
- `pageCtrl` - в данной папке содержатся контроллеры, соединяющие модель данных и представление;
- `utility` - в данной папки располагаются классы-помощники, например: преобразователь строки в дату и наоборот; классы, отвечающие за соединение со сторонними серверами по http-протоколу.
- `data`
 - `comparator` - каталог, в котором создаются классы для настройки сортировок сущностей по свойствам;
 - `dao` - в данном каталоге располагаются классы, позволяющие соединяться с базой данных для выполнения CRUD-операций;
 - `entity` - в каталоге расположены сущности, соответствующими таблицам базе данных.

3.4 Описание функций

Для разработки модулей: авторизация, зачетная книжка, журнал старосты и посещаемость будет использовано схема базы данных, представленная на рисунке 4. Более востребованные таблицы представлены в приложение А.

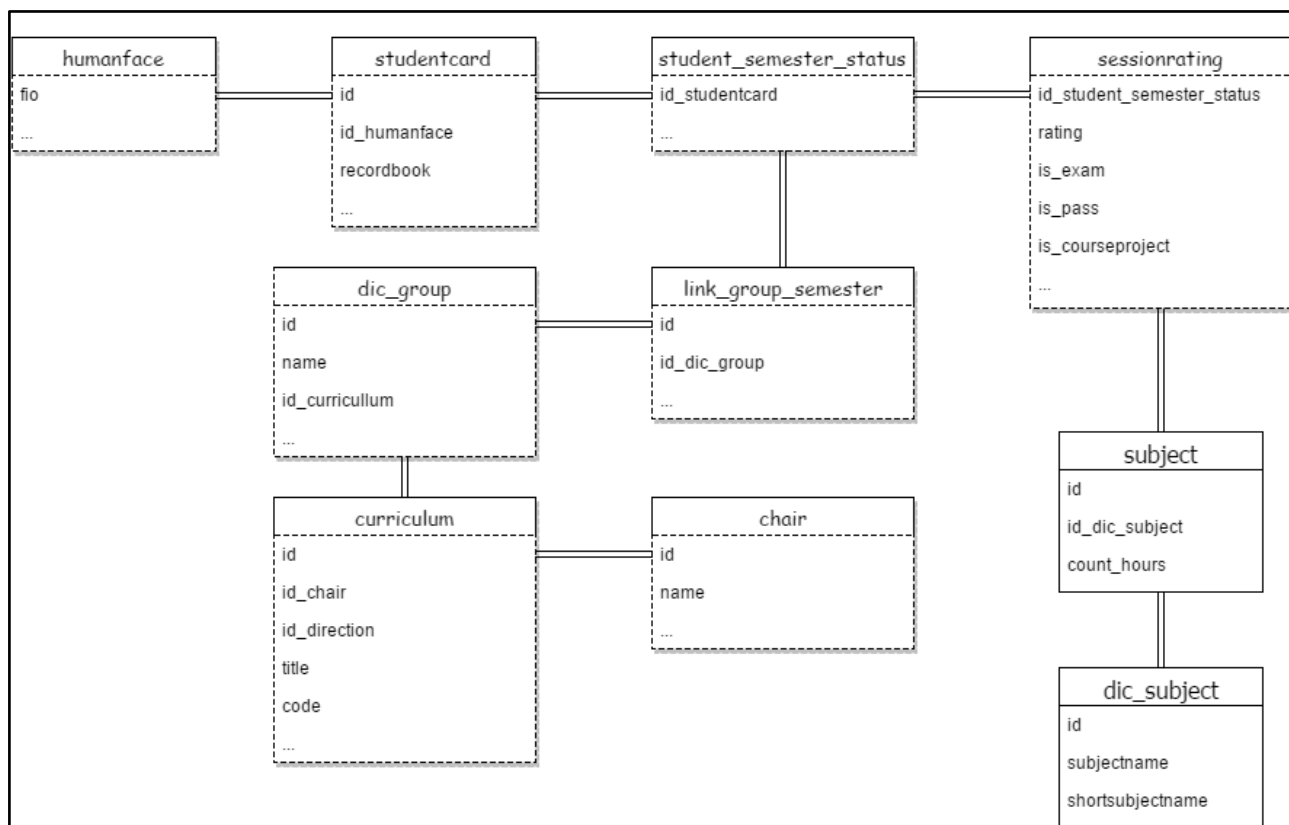


Рисунок 4 - схема базы данных СУБД PostgreSQL

Авторизация

Для входа в систему необходима авторизация. Логин и пароль для входа в систему являются данные LDAP, использующиеся для подсоединения к WI-FI или другим электронным ресурсам СФУ. Окно авторизации представлено на рисунке 5.

Алгоритм авторизации:

- 1) вводится логин и пароль пользователя и нажимается кнопка “Ок”;
- 2) введенные данные отправляются на LDAP сервер, а он в свою очередь отправляет обратно информацию об успешной или не успешной аутентификации;
- 3) если аутентификация прошла успешно, то проверяется, имеется ли данный пользователь у нас в системе;
- 4) если пользователь существует в системе, то проверяется, имеется ли студенческая карточка у пользователя (разграничение прав на

использования модуля: могут пользоваться только студенты), если карточка имеется, то авторизация прошла успешно.

Рисунок 5 - авторизация

Модуль “Зачетная книжка”

Данный модуль предназначен для представления информации по промежуточной аттестации студента. Основное окно представлено на рисунке 6.

Курс 4						
2015-2016 / весенний семестр						
#	Предмет	Кол-во часов	Оценка	Дата	Преподаватель	Примечание
1	Основы построения трансляторов	180 (5 з.е.)	Отлично	2016-05-04	Кузнецов Александр Сергеевич	
2	Системы искусственного интеллекта	144 (4 з.е.)	-		Пенькова Татьяна Геннадьевна	Ведомость не сдана в деканат
3	Междисциплинарный курсовой проект	108 (3 з.е.)	-		Михалев Антон Сергеевич	Ведомость не сдана в деканат
4	Основы правовых знаний	108 (3 з.е.)	Зачет	2016-04-28	Громыко Вячеслав Александрович	
5	Преддипломная практика	108 (3 з.е.)	-			
6	Проектирование человеко-машинного интерфейса	72 (2 з.е.)	Зачет	2016-04-29	Шелопин Александр Сергеевич	
7	Экономика программной инженерии	108 (3 з.е.)	Зачет	2016-04-26	Евдокимов Иван Валерьевич	
8	Междисциплинарный курсовой проект	108 (3 з.е.)	-		Михалев Антон Сергеевич	Ведомость не сдана в деканат
9	Основы построения трансляторов	180 (5 з.е.)	-		Михалев Антон Сергеевич	

Экзамен
Зачет
Курсовой проект
Курсовая работа
Практика

Рисунок 6 - модуль “Зачетная книжка”

В таблице доступно 6 колонок: предмет, кол-во часов, оценка, дата (дата занесения оценки в ведомость), преподаватель (ФИО преподавателя) и примечание. Наиболее интересная колонка, с точки зрения функционала, является примечание.

Колонка примечание имеет несколько вариантов представления:

- 1) Если сессия еще не началась, то показывается дата консультации и дата сдачи предмета, если они занесены в БД.
- 2) Если дата сдачи предмета уже прошла, а оценки нету, то это означает, что преподаватель не подписал оценки в АСУ ИКИТ. В данном случае в колонке примечания будет представлена запись: “Ведомость не сдана в деканат”.
- 3) Если студент, не сдал предмет, то ему будет назначена пересдача. Информация с датой пересдачи будет также представлена в колонке примечаний.

Модуль “Журнал посещаемости”

Журнал посещаемости предназначен для старост групп, которые будут заносить информацию о посещении студентов группы в систему. Данный модуль представлен на рисунке 7.

03.05.16

?

#	Студент	Проект	Эконом	Основы	Основы
1	Беккер Владимир Андреевич	н	н	н	
2	Гасенко Владислав Сергеевич			н	
3	Джабиев Владислав Рафаэлович	н	н	н	н
4	Димухаметов Максим Олегович	н			
5	Копылов Иван Владимирович				
6	Костюкова Анастасия Николаевна				
7	Непомнящих Вячеслав Федорович		н		н
8	Пепуль Андрей Валентинович	н	н	н	
9	Полежаева Ольга Андреевна				
10	Пятов Никита Евгеньевич		н	н	н
11	Судаков Илья Игоревич	н			н
12	Тарасов Дмитрий Алексеевич				
13	Финошкин Алексей Иванович		н		
14	Хомусьяк Денис Игоревич				
15	Шлак Галина Владимировна				
16	Яковлев Юрий Валерьевич	н		н	

Дисциплина

Проектирование человеко-машинного интерфейса

Основы правовых знаний

Основы построения трансляторов

Экономика программной инженерии

Преддипломная практика

Системы искусственного интеллекта

Междисциплинарный курсовой проект

Рисунок 7 - модуль “Журнал посещаемости”

В модуле журнал посещаемости предоставляется возможность выбрать день, по которому нужно заполнить посещаемость. В правом списке дисциплин нужно выбрать нужные предметы в том порядке, в котором они стоят в расписании, они добавятся в левый список. Далее в левом списке по нажатию на клетку пересечения нужной фамилии и предмета, ставится или отменяется присутствие человека отметкой “н”.

Модуль “Посещаемость”

Данный модуль показывает студенту его посещаемость за текущий семестр. Студенту предоставляется возможность выбрать месяц, по которому он может наблюдать свою посещаемость за выбранный месяц. Данный интерфейс представлен на рисунке 8.

Семестр: весна 2015-2016
Начало семестра: 08.02.2016
Конец семестра: 01.05.2016

Предыдущий

Май

Следующий

ПН	ВТ	СР	ЧТ	ПТ	СБ	ВС
25	26	27	28	29	30	01
02	03	04	05	06	07	08
09	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31	01	02	03	04	05
ПН	ВТ	СР	ЧТ	ПТ	СБ	ВС

Занятий не было/староста не заполнил журнал посещаемости.

Посещено часть занятий.

100%-ая посещаемость.

Рисунок 8 - модуль “Посещаемость”

Чтобы узнать подробную посещаемость за день, нужно нажать на конкретный день. Откроется модальное окно, на котором будет представлена нужна информация. Данное окно представлено на рисунке 9.

Посещаемость за 10-05-2016

✕

#	Предмет
1	Системы искусственного интеллекта
2	Системы искусственного интеллекта
3	Системы искусственного интеллекта
4	Системы искусственного интеллекта

Рисунок 9 - посещаемость за определенный день

Модуль “Уведомления”

Данный модуль предназначен для уведомления студентов по следующим темам:

- уведомление о назначении пересдачи;
- уведомление о назначении комиссии;
- персональное уведомление, написанное в АИС Деканат сотрудником УОО.

При входе в “Личный кабинет студента”, в левом меню есть кнопка “Сообщения”, при наличии новых уведомлений, будет указано их количество. Модуль “Уведомления” представлен на рисунке 10.

Личный кабинет студента	
Профиль	
Зачетная книжка	
Сообщения	
Опросник	
Посещаемость	
История обучения	
Список приказов	
Сервисы СФУ	
Выйти	






#	...	Тема	Сообщение	Дата
1		Здоровенный текст	Какой же здоровенный текст в оповещениях ...	06.04.2016
2		Очередной тест после заливки	Тестовое сообщение ...	21.01.2016
3		Перезалив	Максим, перезалив прошел успешно? ...	18.01.2016
4		ALARM!!!	ALARM!!!! ...	18.01.2016
5		Первое сообщение	Тестовое сообщение, специально написанное для...	18.01.2016

Рисунок 10 - модуль “Уведомления”

Для текущего модуля на стороне InterSystems Cache был разработан REST-сервис, диаграмма данного сервиса представлена на рисунке 11.

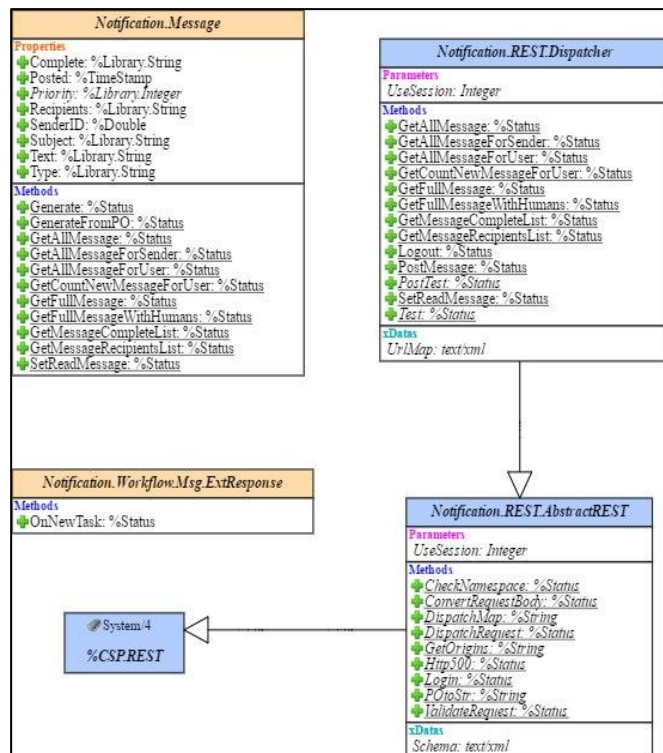


Рисунок 11 - диаграмма классов модуля “Уведомления” на REST-сервисе

Модуль “Опросник”

Данный модуль предназначен для удобного взаимодействия УОО со студентами института. Блоки вопросов заполняют сотрудники УОО в АИС ИКИТ, после чего добавляют группы или конкретных студентов, которым хотят отправить составленный опрос. Далее составленный опрос отправляется на REST-сервис InterSystems Cache. Для данного сервиса была разработана диаграмма классов, представленная на рисунке 12.

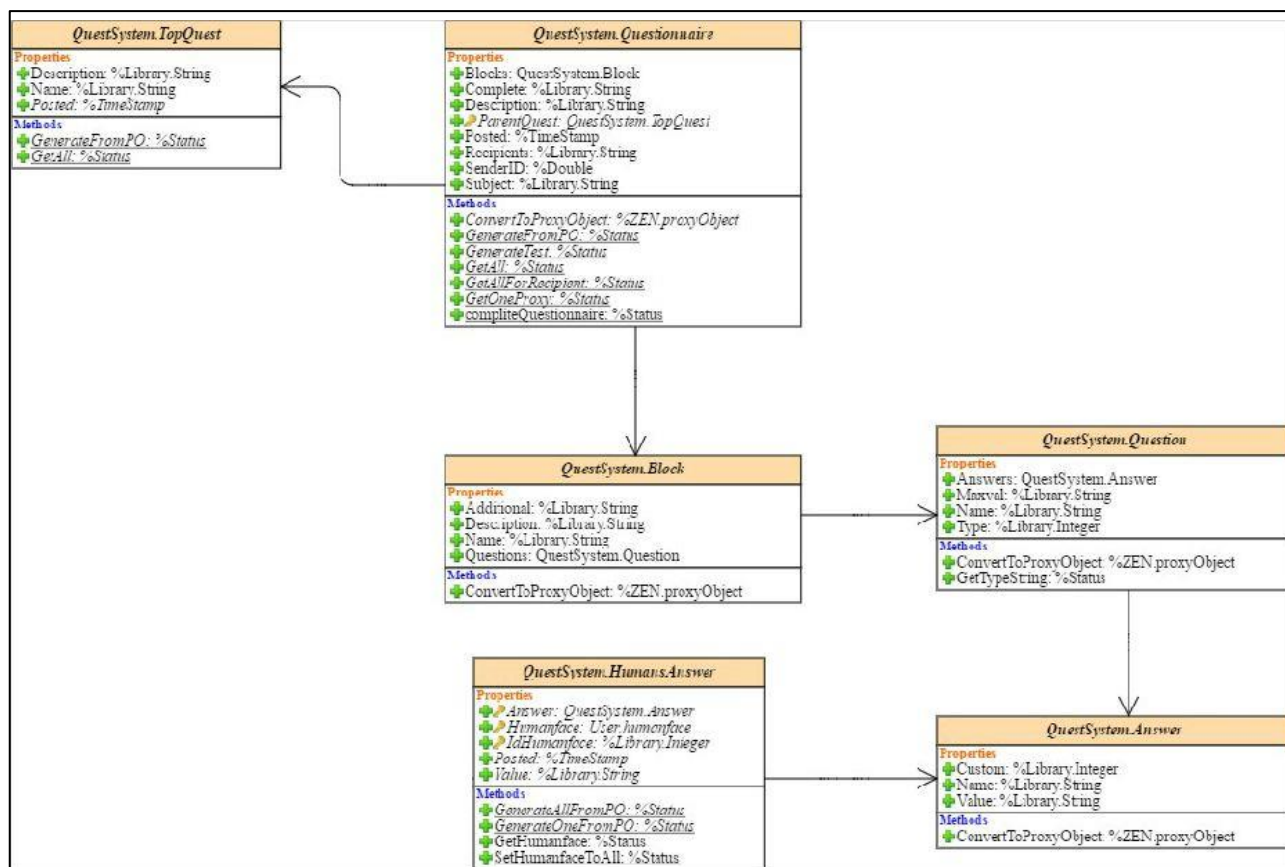


Рисунок 12 - диаграмма классов модуля “Опросник” на REST-сервисе

После того, как отправили на REST-сервис опрос, он стал доступен студентам. Доступные опросы можно посмотреть в модуле “Опросник”. При выборе интересующего нас опроса, нажимаем на него, открывается диалоговое окно для дальнейшего прохождения. При ответе на вопрос, помечается ответ в навигационном меню и в меню контента блока. После того, как ответят на все вопросы разблокируется кнопка “Отправить ответы”. По её нажатию отправятся все ответы на сервер REST-сервис. Пример опроса представлен на рисунке 13.

Личный кабинет

Тема опросника

Описание опросника

Отправить ответы

Защита информации

Полнота и качество электронног...

Доступность изложения материал...

Корректное отношение преподава...

Корректное отношение преподава...

Добросовестное отношение препода...

Объективность выставляемых оце...

Модели и методы искусственного интеллекта

Полнота и качество электронног...

Доступность изложения материал...

Корректное отношение преподава...

Корректное отношение преподава...

Защита информации

Полнота и качество электронного образовательного ресурса

1 2 3 4 5

Доступность изложения материала преподавателем

1 2 3 4 5

Корректное отношение преподавателя к студентам

1 2 3 4 5

Корректное отношение преподавателя к студентам

1 2 3 4 5

Рисунок 13 - модуль “Опросник”

Описание модулей

Для удобства пользователей был разработан данный модуль. Он предназначен для краткого пользовательского руководства использования модуля. Перейти к данному модулю можно нажав на кнопку “?” , находящийся в правом верхнем углу. Оглавление данного модуля представлено на рисунке 14.

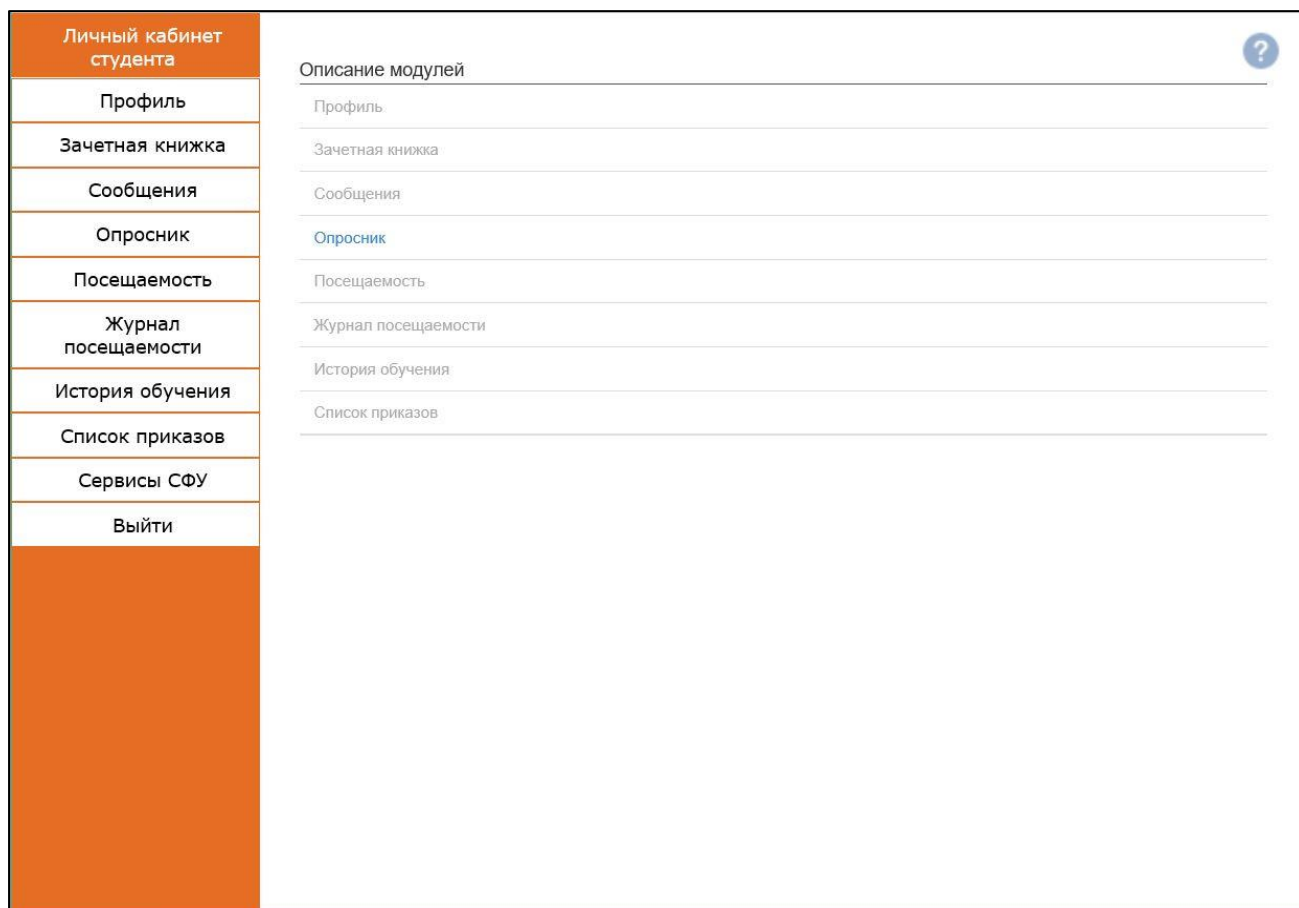


Рисунок 14 - описание модулей

Пример описание модуля Опросник представлен на рисунке 15. Данный раздел показывает все окна модуля и его функционал.

Описание "Опросника"

- Список опросников находится при переходе в модуль "Опросник", где расположен список опросников доступный вам.

#	Тема	Отправитель	Дата
1	Анкета студента (осень 2015) - КИ12-18Б	Сомова М. В.	2016-03-23

- Перед вами откроется окно с опросником, который вы выбрали:

Тема опросника

Описание опросника

Отправить ответы

Защита информации

Полнота и качество электронног...

Доступность изложения материал...

Корректное отношение преподава...

Корректное отношение преподава...

Добросовестное отношение препода...

Объективность выставяемых оце...

Модели и методы искусственного интеллекта

Полнота и качество электронног...

Доступность изложения материал...

Корректное отношение преподава...

Корректное отношение преподавателя к студентам

1 2 3 4 5

Корректное отношение преподавателя к студентам

1 2 3 4 5

Добросовестное отношение преподавателя (отсутствие опоздания, пропусков, хороший уровень подготовки к занятиям)

1 2 3 4 5

Объективность выставяемых оценок

- Под цифрой 1 показано только что отвеченный вопрос. Отображение отвеченного ответа проходит, как в зоне ответа, так и в навигационном меню.
- Под цифрой 2 показано кнопка "Отправить ответы", она будет заблокировано до тех пор, пока не ответите на все вопросы.
- Под цифрой 3 кнопка "Описание опросника", которая открывает "легенду", на которой показано краткое описание опросника.
- P.S.: отвечать нужно все в одной сессии, т.к. ответы не сохраняются

Рисунок 15 - описание модуля "Опросник"

Заключение

В результате работы над проектом была создан и апробирован личный кабинет студента СФУ ИКИТ, в виде следующих модулей:

- зачетная книжка;
- журнал старосты;
- модуль посещаемости;
- оповещения;
- опросник.

Данный проект был внедрен в эксплуатацию с января 2016 года в Институт космических и информационных технологий Сибирского федерального университета. Были выявлены преимущества, недостатки и планы развития системы.

Преимущества системы:

- информация о назначении пересдач сразу попадает в кабинет студента в модуль оповещения;
- студенты могут отслеживать оценки, которые проставил им преподаватель в ведомость;
- бумажный журнал старосты заменен на электронный;
- создание опросов для студентов по учебно-организационным вопросам.

Недостатки системы:

- отсутствие динамического сохранения результатов опроса;
- в модуле оповещения автоматически отправляется информация только информация о пересдачах.

Планы развития системы:

- разработка модуля расписания;
- адаптация интерфейса под мобильные устройства.

Список литературы

- 1) Открытые системы [Электронный ресурс]
<http://www.osp.ru/os/2004/12/184900/> (Дата обращения 12.06.2016).
- 2) Лаборатория математического моделирования и информационных систем (ММИС) [Электронный ресурс] <http://www.mmis.ru/Default.aspx?tabid=166> (Дата обращения 12.06.2016).
- 3) Абитуриент СФУ [Электронный ресурс] <http://abiturient.sfu-kras.ru/> (Дата обращения 13.06.2016).
- 4) Энциклопедия языков программирования [Электронный ресурс]
<http://progopedia.ru/language/java/> (Дата обращения 13.06.2016).
- 5) Описания ZK фреймворка [Электронный ресурс]
<http://books.zkoss.org/zkessentials-book/master/> (Дата обращения 14.06.2016).
- 6) Elliott, James. Hibernate: A Developer's Notebook. — 1st ed. — O'Reilly Media, 2004 — 190 p. [Текст].
- 7) Википедия [Электронный ресурс] https://ru.wikipedia.org/wiki/Система_управления_базами_данных (Дата обращения 14.06.2016).
- 8) Информационный сайт о PostgreSQL [Электронный ресурс]
<http://postgresql.ru.net/manual/intro-what-is.html> (Дата обращения 14.06.2016).
- 9) Официальный сайт IntersSystems [Электронный ресурс]
<http://www.intersystems.com/ru/our-products/cache/cache-overview/> (Дата обращения 14.06.2016).
- 10) Официальная книга по Subversion [Электронный ресурс] <http://svnbook.red-bean.com/nightly/ru/svn-book.html> (Дата обращения 15.06.2016).
- 11) Википедия [Электронный ресурс] https://ru.wikipedia.org/wiki/Интегрированная_среда_разработки (Дата обращения 15.06.2016).
- 12) Официальный сайт компании JetBrains [Электронный ресурс]
<https://www.jetbrains.com/idea/> (Дата обращения 15.06.2016).
- 13) Информационный сайт про Apache Maven [Электронный ресурс]
<http://www.apache-maven.ru/> (Дата обращения 15.06.2016).

ПРИЛОЖЕНИЕ А

Схема СУБД PostgreSQL

HUMANFACE	
Таблица, содержащая основные сведения о человеке, независимо от его роли.	
birthday	Дата рождения
passportdateofgrant	Дата выдачи паспорта
is_active	Признак того, что запись активна (не удалена)
sex	Пол: 0 - женский, 1 - мужской
dateofaddition	Дата добавления в БД
retierinsure	
id_systemuser	Ссылка на системного пользователя, который создал запись
family	Фамилия
name	Имя
patronymic	Отчество
passportserial	Серия паспорта
passportnumber	Номер паспорта
passportwhogrant	Кем выдан паспорт
email	Адрес электронной почты
inn	ИНН
nationality	Национальность

STUDENTCARD	
Запись о студенте, единственная на протяжении всего обучения, даже если переводится между институтами.	
id_humanface	Ссылка на соответствующую запись в таблице humanface
studentcategory	
contractdate	Дата договора о платном обучении
recordbook	Номер зачётной книжки
contractnumber	Номер договора о платном обучении
other_dbuid	Идентификатор из внешней БД
other_login	
other_email	Адрес электронной почты
other_esoid	

STUDENT_SEMESTER_STATUS	
Состояние/статус студента в каждом семестре на протяжении обучения.	
id_studentcard	Ссылка на соответствующую запись в таблице studentcard
id_link_group_semester	Ссылка на соответствующую запись в таблице link_group_semester
is_government_financed	Флаг бюджетной формы обучения
is_put_app_for_social_grant	Флаг подачи заявления на социальную стипендию
is_get_social_grant	Флаг начисления (получения) социальной стипендии

is_deducted	Флаг того, что студент отчислен в текущем семестре
is_scientificwork	Флаг научной работы
is_publicwork	Флаг общественной работы
is_chernobolec	Флаг того, что студент является ликвидатором аварии на ЧАЭС
is_sirola	Флаг того, что студент сирота
is_invalid	Флаг того, что студент имеет инвалидность
is_sessionprolongation	Флаг того, что студент имеет продление сессии
is_combatants	Флаг того, что студент является участником боевых действий
is_academicleave	Флаг того, что студент в академическом отпуске
is_listener	Слушатель: посещает занятия, но неофициально. Не учитывается в приказах и отчётах
sessionresult	Закрыв ли студент сессию меньше 0 - количество долгов (с отрицательным знаком) 0 - пустой запрос 1 - закрыл, есть 3 2 - закрыл, только на 4 3 - закрыл, на 4 и 5 4 - закрыл, только на 5 По умолчанию отрицательное число, чтобы студенты без оценок не попадали в переводной и стипендиальный приказы.
is_trustagreement	Флаг целевого договора
formofstudy	Форма обучения: 1 - очная, 2 - заочная, 3 - вечерняя
is_seconddegree	Флаг второго высшего образования (для заочников)
is_group_leader	Флаг старосты группы
debtcount	Количество долгов
is_educationcomplete	Флаг того, что студент завершил обучение
prolongationbegindate	Дата начала продления сессии
prolongationenddate	Дата окончания продления сессии
dateofscholarshipbegin	Дата начала выплат стипендии
dateofscholarshipend	Дата окончания выплат стипендии
is_ord_scholarship	Служебное поле для формирования приказов
is_ord_transfer	Служебное поле для формирования приказов

LINK_GROUP_SEMESTER	
Связь между группой и семестром. Новая запись создается для каждой группы в каждом семестре.	
id_semester	Ссылка на соответствующую запись в таблице semester
id_dic_group	Ссылка на родительскую группу
dateofbeginsemester	Дата начала семестра
dateofendsemester	Дата окончания семестра
dateofbeginsession	Дата начала сессии
dateofendsession	Дата окончания сессии

dateofbeginpassweek	Дата начала зачётной недели
dateofendpassweek	Дата окончания зачётной недели
dateofbeginvacation	Дата начала каникул
dateofendvacation	Дата окончания каникул
dateofbegingrant	Дата начала выплаты стипендии
dateofendgrant	Дата окончания выплаты стипендии
semesternumber	Номер семестра для группы
course	Номер курса

DIC_GROUP	
Студенческая группа. Одна запись на протяжении всего срока обучения.	
id_institute	Связь с институтом которому принадлежит группа
id_curriculum	Ссылка на учебный план группы
parent_group	Родительская группа (при разбиении на подгруппы в процессе обучения)
dateofbegin	Дата (год) формирования группы
dateofend	Дата (год) расформирования группы
is_military	Студенты УВЦ
is_active	Признак того, что запись активна (не удалена)
groupname	Наименование учебной группы

SEMESTER	
Семестры. Одна запись в данной таблице отвечает за 1 семестр	
id_schoolyear	Ссылка на таблицу Учебный год
id_institute	Ссылка на таблицу Институт
formofstudy	Форма обучения: 1 - очная, 2 - заочная
is_current_sem	Флаг того, что это текущий семестр для выбранного института и формы обучения
is_closed	Флаг того, что семестр закрыт, запрещено редактирование данных за этот семестр

CURRICULUM	
Учебный план	
id_direction	Ссылка на направление
formofstudy	Форма обучения: 1 - очная, 2 - заочная
distancetype	Заочная форма обучения: 1 - заочно-очная, 2 - заочная
periodofstudy	Период обучения, может быть дробным числом
qualification	Квалификация: 1 - инженер, 2 - бакалавр, 3 – магистр
enteryear	Год набора по данному учебному плану
generation	Поколение учебного плана
id_chair	Идентификатор кафедры, за которой закреплён учебный план
specialitytitle	Наименование специальности
directioncode	Код направления
qualificationcode	Код квалификации
programcode	Код программы
planfilename	Название файла с учебным планом для привязки к планам в системе

SESSIONRATING	
Итоговая успеваемость студента	
id_student_semester_status	Ссылка на студента
is_exam	Флаг экзамена
is_pass	Флаг зачёта
is_courseproject	Флаг курсового проекта
is_coursework	Флаг курсовой работы
examrating	Оценка за экзамен (учитывать совместно с флагом is_exam)
passrating	Оценка за зачёт (учитывать совместно с флагом is_pass): 0 - не зачтено, 1 – зачтено
courseprojectrating	Оценка за курсовой проект (учитывать совместно с флагом is_courseproject)
courseworkrating	Оценка за курсовую работу (учитывать совместно с флагом is_coursework)
esogradecurrent	Текущее количество баллов по электронным обучающим курсам, которое набрано студентом к текущему моменту
esogrademax	Максимальное количество баллов по электронным обучающим курсам, которое можно набрать к текущему моменту
is_esostudy	Флаг обучения по электронным обучающим курсам (если для дисциплины указан обучающий курс)
skipcount	Количество пропусков занятий на текущий момент (из электронного журнала)
visitcount	Количество посещений занятий на текущий момент (из электронного журнала)
id_subject	Ссылка на дисциплину
is_practic	Флаг практики (летней, учебной, производственной и т.д.)
practicrating	Оценка за практику (учитывать совместно с флагом is_practic)
esoexamrating	Оценка за экзамен из электронного журнала
esopassrating	Оценка за зачёт из электронного журнала
esocourseprojectrating	Оценка за курсовой проект из электронного журнала
esocourseworkrating	Оценка за курсовую работу из электронного журнала
esocourseprojecttheme	Тема курсового проекта из электронного журнала
esocourseworktheme	Тема курсовой работы из электронного журнала
esotmpexamtutor	Временное поле с ФИО преподавателя, выставившего итоговую оценку
esotmppasstutor	Временное поле с ФИО преподавателя, выставившего итоговую оценку
esotmpcourseprojecttutor	Временное поле с ФИО преподавателя, выставившего итоговую оценку
esotmpcourseworktutor	Временное поле с ФИО преподавателя, выставившего итоговую оценку

SESSIONRATINGHISTORY	
Таблица со списком изменений в успеваемости	
id_sessionrating	Ссылка на родительскую запись в таблице успеваемости
is_exam	Флаг экзамена
is_pass	Флаг зачёта
is_courseproject	Флаг курсового проекта
is_coursework	Флаг курсовой работы
newrating	Новая оценка
oldrating	Старая оценка
changedatetime	Дата и время обновления оценки
id_systemuser	Сотрудник обновивший оценку
is_practic	Флаг практики

SUBJECT	
Таблица с дисциплинами учебного плана	
id_curriculum	Ссылка на учебный план
id_dic_subject	Ссылка на наименование дисциплины в справочнике
id_chair	Ссылка на кафедру, отвечающую за дисциплину
parent_subject	Ссылка на запись из предыдущего семестра, если дисциплина разбивается по семестрам
hourscount	Количество часов "Всего"
hoursaudcount	Количество часов "Всего аудиторных"
is_exam	Флаг экзамена
is_pass	Флаг зачёта
is_courseproject	Флаг курсового проекта
is_coursework	Флаг курсовой работы
is_practic	Флаг практики (летней, учебной, производственной и т.д.)
is_active	Признак того, что запись активна (не удалена)
subjectcode	Блок дисциплины из учебного плана
hourslection	(Лек) Часы на лекцию
hourslabor	(Лаб) Часы на лабораторные работы
hourspractic	(Пр) Часы на практические занятия
semesternumber	Номер семестра

DIC_SUBJECT	
Справочник названий дисциплин	
subjectshortname	Короткое наименование дисциплины
subjectname	Полное наименование дисциплины

CHAIR	
Таблица кафедр	
parent_chair	Указатель на родительскую кафедру (для дочернего подразделения, например, НУЛа)

DEPARTMENT	
Таблица с подразделениями (кафедры, деканаты и т.д.). Родительская таблица.	
id_institute	Ссылка на институт

id_chair	Ссылка на кафедру, если подразделение является кафедрой
id_deanery	Ссылка на деканат, если подразделение является деканатом
departmenttype	Строковая запись типа подразделения
is_active	Признак того, что запись активна (не удалена)
fulltitle	Полное наименование подразделения

ATTENDANCE	
Таблица посещаемости	
id_student_semester_status	Ссылка на студента
id_link_group_semester_subject	Ссылка на связующую таблицу, связывающая группу в семестре и предмет
attend	Флаг того, посетил студент или нет занятие
pos	Показатель на каком месте стоит предмет по расписанию
visitdate	Дата проведения занятия

ПРИЛОЖЕНИЕ Б

Диаграмма классов InterSystems Cache модуля “Опросник” и “Оповещения”

NOTIFICATION.MESSAGE	
Таблица уведомлений	
Complete	Список идентификаторов пользователей, прочитавших данное уведомление
Posted	Дата отправления уведомления с АИС “Деканат”
Priority	Приоритет уведомлений
Recipients	Список пользователей, кому предназначено уведомление
SenderID	Идентификатор отправителя
Subject	Тема уведомления
Text	Текст уведомления
Type	Тип уведомления (оповещение, предупреждение, напоминание и т.д.)

QuestionSystem.TopQuest	
Таблица тематики опросника	
Description	Описание тематики
Name	Название тематики
Posted	Дата создания тематики в АИС “Деканат”

QuestionSystem.Questionnaire	
Таблица опросника	

Blocks	Ссылка на список блоков, присущие опроснику
Complete	Список идентификаторов пользователей, которые ответили на опрос
Description	Описание опросника
ParentQuest	Ссылка на тематику опросника
Posted	Дата создания опросника в АИС “Деканат”
Recipients	Список людей, которым предстоит ответить опрос
SenderID	Идентификатор пользователя, кто создал опрос
Subject	Тема опроса

QuestionSystem.Block	
Таблица блока опроса	
Additional	Дополнительная информация, которая может ссылаться на предмет или преподавателя в СУБД PostgreSQL АИС “Деканат”
Description	Описание блока
Name	Название блока
Questions	Ссылка на список вопросов, присущие блоку

QuestionSystem.Question	
Таблица вопроса	
Answers	Ссылка на список ответов, присущие вопросу
Type	Тип ответов, данного вопроса <ul style="list-style-type: none"> - 0 - один из списка - 1 - множественный список - 2 - текстовый ответ - 3 - диапазон значений
MaxVal	Максимальное значение для типа диапазон значений
Name	Название блока

QuestionSystem.Answer	
Таблица ответа	
Custom	Флаг того, является ли ответ кастомизируемым. Кастомный ответ - это когда у него тип (например, один из списка), а ему вместо выбора из списка нужно дописать текстовый ответ.
Value	Значение ответа
Name	Название блока

QuestionSystem.HumanAnswer	
Таблица, которая связывает ответ и пользователя	
IdHumanface	Идентификатор пользователя
Value	Значение ответа